# Proof Normalization Modulo

Gilles Dowek, Benjamin Werner

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Proof normalization modulo*

Gilles Dowek and Benjamin Werner

**No 3542**

Novembre 1998

THÈME 2


ISSN 0249-6399

# Proof normalization modulo

Gilles Dowek* and Benjamin Werner†

Thème 2 — Génie logiciel
et calcul symbolique
Projet Coq

Rapport de recherche n ° 3542 — Novembre 1998 — 40 pages

**Abstract:** We consider a class of logical formalisms, in which first-order logic is extended by identifying propositions modulo a given congruence. We particularly focus on the case where this congruence is induced by a confluent and terminating rewrite system over the propositions. This extension enhances the power of first-order logic and various formalisms, including higher-order logic, can be described in this framework.

We conjecture that proof normalization and logical consistency always hold over this class of formalisms, provided some minimal conditions over the rewrite system are fulfilled. We prove this conjecture for some subcases, including higher-order logic.

At last, we extend these results to classical sequent calculus.

**Key-words:** deduction modulo, natural deduction, sequent calculus, cut elimination.

*(Résumé : tsvp)*

* Gilles.Dowek@inria.fr http://coq.inria.fr/~dowek/
† Benjamin.Werner@inria.fr http://coq.inria.fr/~werner/

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : 01 39 63 55 11 - International : +33 1 39 63 55 11
Télécopie : (33) 01 39 63 53 30 - International : +33 1 39 63 53 30

# La normalisation des démonstrations modulo

**Résumé :** Nous nous intéressons à une extension de la logique du premier ordre dans laquelle les propositions sont identifiées modulo une certaine congruence et plus particulièrement au cas où cette congruence est définie par un système de réécriture sur les propositions qui est confluent et qui termine. Cette extension augmente la puissance de la logique du premier ordre et divers formalismes, tels la logique d'ordre supérieur peuvent être décrits dans ce cadre.

Nous conjecturons la normalisation des démonstrations et la cohérence logique de ces formalismes, pourvu que le système de réécriture vérifie certaines conditions élémentaires. Nous démontrons cette conjecture dans des cas particulier, entre autres pour la logique d'ordre supérieur.

Enfin, nous étendons ces résultats au calcul des séquents classique.

**Mots-clé :** déduction modulo, déduction naturelle, calcul des séquents, élimination des coupures.

# Introduction

## Motivations

A proof-system implements a given logical formalism. The choice of this formalism is important since in the field of actually mechanically checked formal proofs logical formalisms are required not only to be *expressive* (logical complexity) but also *practicable*. The following issues are critical.

- The conciseness of proofs: in recent practical developments, it clearly appeared that the size of the proof-object and thus its handling and the practicability of proof-checking can become critical; and the formalism in which this proof is expressed is an important factor to that respect.
- A side-effect of the latter is also that smaller proofs often reflect more closely the mathematical intuition. In other words, this allows the user to better grasp the mathematical object he/she produces.
- Last but not least, automatic proof-search and more generally computer-provided user help depend upon the chosen formalism. It is well-known that proof synthesis algorithms are expressed more or less clearly in different logics.

In this respect, a particular attention has often been given to the distinction between *calculation* and *reasoning* steps. Schematically, the first can be unambiguously and mechanically performed and reproduced; whereas the latter correspond to the application of a logical inference rule, whose choice is the responsibility of the author/user. As a consequence, the calculation steps can be omitted in the proof objects. A typical instance is the conversion rule of type theories; a typical application is recent work using computational reflection like, for instance, [1].

## Deduction modulo

*Deduction modulo* is a way to remove computational arguments from proofs by reasoning modulo a congruence on propositions. This idea is certainly not new. For instance, in a language containing an associative binary function symbol $+$, Plotkin [15] proposes to identify propositions such as $P((a+b)+c)$ and $P(a+(b+c))$ that differ only by a rearrangement of brackets. Following Plotkin, Stickel [16] proposes a proof search method where some equational axioms are mixed with the unification algorithm leading to equational unification [6, 11]. Similarly, the *conversion rule* of type theories [13, 2, 14] a.o. identifies propositions w.r.t. generalized $\beta$-reduction: the

propositions $1 + 1 = 2$ and $2 = 2$ are logically identical. In Second order functional arithmetic [12] it is possible to use an equational axiom without recording this step in the proof.

This separation between computations and deductions deserves to be studied for itself, independently of a particular application or a particular formalization of mathematics, i.e. in the most general framework: first-order logic.

The usual way to define a congruence on propositions is to define a congruence on terms, for instance by a set of equations or rewrite rules and to extend this congruence to propositions. However, in [3], a class of logical systems has been introduced, in which the congruence is defined directly over the structure of the whole proposition. A striking point is that adding well-chosen congruences enhances the logical expressivity of the formalism; typically, it leads to a first-order and axiom-free presentation of higher-order logic. A interesting application is that enforcing the distinction between calculation and reasoning leads to a very nice clarification of higher-order resolution. See [3] for details.

### About this work

In this paper, we study deduction modulo from the proof-theoretic viewpoint and more particularly the properties of cut elimination and consistency. Proof normalization for such proof systems does not always hold and we present several counter-examples below; but we conjecture that proofs always normalize for congruences that can be defined by a confluent and terminating rewrite system, which rewrites terms to terms and atomic propositions to arbitrary ones.

In this paper we show some particular cases of this conjecture: we show that proof normalization holds for our presentation of higher-order logic, for all congruences defined by a confluent and terminating rewriting system rewriting terms to terms and atomic propositions to *quantifier free* propositions and for *positive* rewrite systems i.e. ones having rewriting terms to terms and atomic propositions to propositions without the symbol $\Rightarrow$.

## 1 Deduction modulo

### 1.1 Natural deduction modulo and sequent calculus modulo

In deduction modulo, the notions of language, term, proposition are that of (many-sorted) first-order logic [5, 7]. The definitions below are well-known and thus not too detailed.

We consider a, at most, numerable set of *sorts*, whose elements will be denoted by $(s, s', s_1 \ldots)$. We consider a numerable set of *variables* of each sort. We give ourselves a set of function symbols and of predicate symbols. Each of these comes with its *rank*. The formation rules for objects and propositions are the usual ones.

- Variables of sort $s$ are terms of sort $s$.
- If $f$ is a function symbol of rank $(s_1, \ldots, s_n, s')$ and $t_1, \ldots, t_n$ are respectively objects of sort $s_1, \ldots, s_n$, then $f(t_1, \ldots, t_n)$ is a well-formed object of sort $s'$.
- If $P$ is a predicate symbol of rank $(s_1, \ldots, s_n)$ and $t_1, \ldots, t_n$ are respectively objects of sort $s_1, \ldots, s_n$, then $P(t_1, \ldots, t_n)$ is a well-formed *atomic proposition*.

Well-formed propositions are built-up from atomic propositions, from the usual connectors and quantifiers $\Rightarrow, \vee, \wedge, \bot, \forall$ and $\exists$. Remark that, implicitly, quantification in $\forall x^s P$ or $\exists x^s P$ is restricted over the sort $s$. As usual, we assume that various relations are decidable (equality over variables, the sort of variables, the rank of symbols, etc).

What characterizes deduction modulo is that a *theory* is formed by a set of axioms $\Gamma$ and a congruence $\equiv$. Figure 1 gives the rules of natural deduction modulo. Figure 2 gives the rules of sequent calculus modulo. As usual, intuitionistic natural deduction is obtained by dropping the excluded middle rule and intuitionistic sequent calculus by restricting to sequents with at most one conclusion.

If the congruence $\equiv$ is decidable, then proof checking is decidable, since we provided the necessary information in the quantifier rules.

**Proposition 1.1** *(Equivalence) For every congruence $\equiv$ there is a theory $\mathcal{T}$ such that $\Gamma \vdash_{\equiv} P$ if and only if $\mathcal{T}\Gamma \vdash P$.*

*Proof.* See [3]. □

The framework we have defined up to here is extremely general. In the following, and to study proof-theoretic properties, we mainly deal with the case where the congruence $\equiv$ is generated by a rewriting relation. The definition is straight-forward.

**Definition 1.1** *We say that a congruence $\equiv$ is defined by a confluent and terminating rewriting system $\mathcal{R}$ rewriting terms to terms and atomic propositions to arbitrary ones when $P \equiv Q$ if and only if $P$ and $Q$ have the same normal form for the system $\mathcal{R}$.*

$$\frac{}{\Gamma \vdash_{\equiv} B}\ \text{axiom if } A \in \Gamma \text{ and } A \equiv B$$

$$\frac{\Gamma, A \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \Rightarrow\text{-intro if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} C \quad \Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} B}\ \Rightarrow\text{-elim if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} A \quad \Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \wedge\text{-intro if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} A}\ \wedge\text{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} B}\ \wedge\text{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} C}\ \vee\text{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \vee\text{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} D \quad \Gamma, A \vdash_{\equiv} C \quad \Gamma, B \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} C}\ \vee\text{-elim if } D \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} A}\ \bot\text{-elim if } B \equiv \bot$$

$$\frac{\Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} B}\ (x, A)\ \forall\text{-intro if } B \equiv (\forall x\ A) \text{ and } x \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ (x, A, t)\ \forall\text{-elim if } B \equiv (\forall x\ A) \text{ and } C \equiv [t/x]A$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} B}\ (x, A, t)\ \exists\text{-intro if } B \equiv (\exists x\ A) \text{ and } C \equiv [t/x]A$$

$$\frac{\Gamma \vdash_{\equiv} C \quad \Gamma, A \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} B}\ (x, A)\ \exists\text{-elim if } C \equiv (\exists x\ A) \text{ and } x \notin FV(\Gamma B)$$

$$\frac{}{\Gamma \vdash_{\equiv} A}\ B \text{ excluded middle if } A \equiv (B \vee (B \Rightarrow \bot))$$

Figure 1: Natural deduction modulo

$$\frac{}{A \vdash_{\equiv} B}\ \text{axiom if } A \equiv B$$

$$\frac{\Gamma, A \vdash_{\equiv} \Delta \quad \Gamma \vdash_{\equiv} B, \Delta}{\Gamma \vdash_{\equiv} \Delta}\ \text{cut if } A \equiv B$$

$$\frac{\Gamma, B_1, B_2 \vdash_{\equiv} \Delta}{\Gamma, A \vdash_{\equiv} \Delta}\ \text{contr-left if } A \equiv B_1 \equiv B_2$$

$$\frac{\Gamma \vdash_{\equiv} B_1, B_2, \Delta}{\Gamma \vdash_{\equiv} A, \Delta}\ \text{contr-right if } A \equiv B_1 \equiv B_2$$

$$\frac{\Gamma \vdash_{\equiv} \Delta}{\Gamma, A \vdash_{\equiv} \Delta}\ \text{weak-left}$$

$$\frac{\Gamma \vdash_{\equiv} \Delta}{\Gamma \vdash_{\equiv} A, \Delta}\ \text{weak-right}$$

$$\frac{\Gamma \vdash_{\equiv} A, \Delta \quad \Gamma, B \vdash_{\equiv} \Delta}{\Gamma, C \vdash_{\equiv} \Delta}\ \Rightarrow\text{-left if } C \equiv (A \Rightarrow B)$$

$$\frac{A\Gamma \vdash_{\equiv} B, \Delta}{\Gamma \vdash_{\equiv} C, \Delta}\ \Rightarrow\text{-right if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma, A, B \vdash_{\equiv} \Delta}{\Gamma, C \vdash_{\equiv} \Delta}\ \wedge\text{-left if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} A, \Delta \quad \Gamma \vdash_{\equiv} B, \Delta}{\Gamma \vdash_{\equiv} C, \Delta}\ \wedge\text{-right if } C \equiv (A \wedge B)$$

$$\frac{\Gamma, A \vdash_{\equiv} \Delta \quad \Gamma, B \vdash_{\equiv} \Delta}{\Gamma, C \vdash_{\equiv} \Delta}\ \vee\text{-left if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} A, B, \Delta}{\Gamma \vdash_{\equiv} C, \Delta}\ \vee\text{-right if } C \equiv (A \vee B)$$

$$\frac{}{\Gamma, A \vdash_{\equiv} \Delta}\ \bot\text{-left if } A \equiv \bot$$

$$\frac{\Gamma, [t/x]A \vdash_{\equiv} \Delta}{\Gamma, B \vdash_{\equiv} \Delta}\ (x, A, t)\ \forall\text{-left if } B \equiv (\forall x\ A)$$

$$\frac{\Gamma \vdash_{\equiv} A, \Delta}{\Gamma \vdash_{\equiv} B, \Delta}\ (x, A)\ \forall\text{-right if } B \equiv (\forall x\ A) \text{ and } x \notin FV(\Gamma\Delta)$$

$$\frac{\Gamma, A \vdash_{\equiv} \Delta}{\Gamma, B \vdash_{\equiv} \Delta}\ (x, A)\ \exists\text{-left if } B \equiv (\exists x\ A) \text{ and } x \notin FV(\Gamma\Delta)$$

$$\frac{\Gamma \vdash_{\equiv} [t/x]A, \Delta}{\Gamma \vdash_{\equiv} B, \Delta}\ (x, A, t)\ \exists\text{-right if } B \equiv (\exists x\ A)$$

Figure 2: Sequent calculus modulo

In this case, the congruence $\equiv$ is decidable.
*Remark:* The definition above can be slightly generalized allowing non-oriented equations relating terms to terms and atomic propositions to atomic propositions (for instance commutativity). See [3] for more details.

## 1.2 Examples

*Example:* (Simplification)

In an integral ring, we can use the usual simplification rules over objects like $a \times 0 \to 0, a \times (b + c) \to a \times b + a \times c$, etc. But we can also add the following rule for simplifying equalities:

$$a \times b = 0 \to a = 0 \vee b = 0$$

or the rule

$$a \times b = a \times c \to a = 0 \vee b = c$$

*Example:* (Higher-order logic)

As mentioned, deduction modulo allows to capture formalisms which go beyond the usual field of first-order logic; here is a faithful encoding of intentional higher-order logic.

The *sorts* are *simple types* inductively defined by

- $\iota$ and $o$ are simple types,
- if $T$ and $U$ are simple types then $T \to U$ is a simple type.

The language is composed of the individual symbols

- $S_{T,U,V}$ of sort $(T \to U \to V) \to (T \to U) \to T \to V$,
- $K_{T,U}$ of sort $T \to U \to T$,
- $\dot{\Rightarrow}$, $\dot{\wedge}$ and $\dot{\vee}$ of sort $o \to o \to o$, $\dot{\bot}$ of sort $o$,
- $\dot{\forall}_T$ and $\dot{\exists}_T$ of sort $(T \to o) \to o$,

the function symbols

- $\alpha_{T,U}$ of rank $(T \to U, T, U)$,

and the predicate symbol

- $\varepsilon$ of rank $(o)$.

As can be guessed, $S_{T,U,V}$ and $K_{T,U}$ are typed combinators and used to express functions. The objects $\dot{\Rightarrow}, \dot{\wedge}, \dot{\vee}, \dot{\bot}, \dot{\forall}_T$ and $\dot{\exists}_T$ allow to represent propositions as objects of sort $o$. Finally, the predicate $\varepsilon$ allows to transform such an object $t$ of type $o$ into the actual corresponding proposition $\varepsilon(t)$. This typical reflection operation appears clearly in the rewrite rules.

$$\begin{aligned}
\alpha(\alpha(\alpha(S,x),y),z) &\to \alpha(\alpha(x,z),\alpha(y,z)) \\
\alpha(\alpha(K,x),y) &\to x \\
\\
\varepsilon(\alpha(\alpha(\dot{\Rightarrow},x),y)) &\to \varepsilon(x) \Rightarrow \varepsilon(y) \\
\varepsilon(\alpha(\alpha(\dot{\wedge},x),y)) &\to \varepsilon(x) \wedge \varepsilon(y) \\
\varepsilon(\alpha(\alpha(\dot{\vee},x),y)) &\to \varepsilon(x) \vee \varepsilon(y) \\
\varepsilon(\dot{\bot}) &\to \bot \\
\varepsilon(\alpha(\dot{\forall},x)) &\to \forall y\ \varepsilon(\alpha(x,y)) \\
\varepsilon(\alpha(\dot{\exists},x)) &\to \exists y\ \varepsilon(\alpha(x,y))
\end{aligned}$$

# 2 The normalization conjecture

We now turn to the study of cut elimination. We place ourselves in intuitionistic natural deduction. Hence, cut elimination boils down to the normalization property with respect to $\beta$-reduction of some $\lambda$-calculus.

## 2.1 Proof-terms

Following Heyting semantics and Curry-Howard isomorphism we write proofs as $\lambda$-terms typed by propositions of first-order logic. These proof-terms can contain both variables of the first-order language (written $x, y, z, \ldots$) and proof variables (written $\alpha, \beta, \ldots$). Terms of the first-order language are written $t, u, v, \ldots$ while proof-terms are written $\pi, \rho, \ldots$

**Definition 2.1** *(Proofs)*

$$\begin{array}{rl} \pi ::= & \alpha \\ & \mid \lambda\alpha\ \pi \mid (\pi\ \pi') \\ & \mid (\pi, \pi') \mid \mathit{fst}(\pi) \mid \mathit{snd}(\pi) \\ & \mid i(\pi) \mid j(\pi) \mid (\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3) \\ & \mid (\mathit{botelim}\ \pi) \\ & \mid \lambda x\ \pi \mid (\pi\ t) \\ & \mid (t, \pi) \mid (\mathit{exelim}\ \pi\ x\alpha\pi') \end{array}$$

As it is now usual, $\lambda$-abstraction models the $\Rightarrow$-intro and $\forall$-intro rules and application the corresponding elimination rules, the pair construct models the $\wedge$-introduction, etc.

Figure 3 gives the typing rules of this calculus. As can easily be seen, we have a typed $\lambda$-calculus, with dependent products. The only originality is that types are identified modulo $\equiv$.

Obviously, a sequent $A_1, \ldots, A_n \vdash_\equiv B$ is derivable in natural deduction modulo if and only if there is a proof $\pi$ such that the judgment $\alpha_1 : A_1, \ldots, \alpha_n : A_n \vdash_\equiv \pi : B$ is derivable in this system.

*Remark:* An alternative presentation of this type system would use a *conversion rule*:

$$\frac{\Gamma \vdash_\equiv t : A}{\Gamma \vdash_\equiv t : B} \text{ if } A \equiv B$$

## 2.2 Proof reduction rules

As usual, the process of cut elimination is modeled by (generalized) $\beta$-reduction. We consider the contextual closure of the reduction rules given figure 4. We write $\pi \to^1 \pi'$ if $\pi$ reduces in one step to $\pi'$, $\pi \to^+ \pi'$ if $\pi$ reduces in at least one step to $\pi'$, and $\pi \to \pi'$ if $\pi$ reduces in an arbitrary number of steps to $\pi'$.

A proof is said to be *normal* if it contains no redex. It is said to be normalizing if it has a normal form and strongly normalizing if all reduction sequences issued from this proofs are finite.

**Proposition 2.1** *(Subject reduction) If* $\Gamma \vdash \pi : P$ *and* $\pi \to \pi'$ *then* $\Gamma \vdash \pi' : P$.

$$\frac{}{\Gamma \vdash_{\equiv} \alpha : B}\ \text{axiom if } \alpha : A \in \Gamma \text{ and } A \equiv B$$

$$\frac{\Gamma \alpha : A \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} \lambda \alpha\ \pi : C}\ \Rightarrow\text{-intro if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C \quad \Gamma \vdash_{\equiv} \pi' : A}{\Gamma \vdash_{\equiv} (\pi\ \pi') : B}\ \Rightarrow\text{-elim if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : A \quad \Gamma \vdash_{\equiv} \pi' : B}{\Gamma \vdash_{\equiv} (\pi, \pi') : C}\ \wedge\text{-intro if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C}{\Gamma \vdash_{\equiv} \mathit{fst}(\pi) : A}\ \wedge\text{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C}{\Gamma \vdash_{\equiv} \mathit{snd}(\pi) : B}\ \wedge\text{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : A}{\Gamma \vdash_{\equiv} i(\pi) : C}\ \vee\text{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} j(\pi) : C}\ \vee\text{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi_1 : D \quad \Gamma \alpha : A \vdash_{\equiv} \pi_2 : C \quad \Gamma \beta : B \vdash_{\equiv} \pi_3 : C}{\Gamma \vdash_{\equiv} (\delta\ \pi_1\ \alpha \pi_2\ \beta \pi_3) : C}\ \vee\text{-elim if } D \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} (\mathit{botelim}\ \pi) : A}\ \bot\text{-elim if } B \equiv \bot$$

$$\frac{\Gamma \vdash_{\equiv} \pi : A}{\Gamma \vdash_{\equiv} \lambda x\ \pi : B}\ (x, A)\ \forall\text{-intro if } B \equiv (\forall x\ A) \text{ and } x \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} (\pi\ t) : [t/x]A}\ (x, A, t)\ \forall\text{-elim if } B \equiv (\forall x\ A)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : [t/x]A}{\Gamma \vdash_{\equiv} (t, \pi) : B}\ (x, A, t)\ \exists\text{-intro if } B \equiv (\exists x\ A)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C \quad \Gamma \alpha : A \vdash_{\equiv} \pi' : B}{\Gamma \vdash_{\equiv} (\mathit{exelim}\ \pi\ x \alpha \pi') : B}\ (x, A)\ \exists\text{-elim if } C \equiv (\exists x\ A) \text{ and } x \notin FV(\Gamma B)$$

Figure 3: Typing rules

$$
\begin{array}{rcl}
(\lambda\alpha\ \pi_1\ \pi_2) & \to & [\pi_2/\alpha]\pi_1 \\
\mathit{fst}(\pi_1,\pi_2) & \to & \pi_1 \\
\mathit{snd}(\pi_1,\pi_2) & \to & \pi_2 \\
\delta(i(\pi_1),\alpha\pi_2,\beta\pi_3) & \to & [\pi_1/\alpha]\pi_2 \\
\delta(j(\pi_1),\alpha\pi_2,\beta\pi_3) & \to & [\pi_1/\beta]\pi_3 \\
(\lambda x\ \pi\ t) & \to & [t/x]\pi \\
(\mathit{exelim}\ (t,\pi_1)\ \alpha x\pi_2) & \to & [t/x,\pi_1/\alpha]\pi_2
\end{array}
$$

Figure 4: Proof reduction rules

**Theorem 2.1** *Provided $\equiv$ is defined from a rewrite system verifying the conditions of definition 1.1, there is no normal proof of the sequent $\vdash_{\equiv} \bot$.*

*Proof.* A normal closed proof can only end with an introduction rule. Thus, we should have a congruence like $\bot \equiv A \wedge B$ or $\bot \equiv A \vee B$, ... which is impossible. □

## 2.3 Counter-examples to termination

To illustrate the subtil link between the combinatorial properties of the rewrite system $\mathcal{R}$ (termination, confluence,...) and the logical properties of the induced formalism (consistency, cut elimination, ...), we here provide two systems where these properties do not hold.

*Example:* (Russell's paradox)

Consider the following rewriting system

$$R \ \to \ (R \Rightarrow S)$$

Modulo this rewriting system, the proof $\lambda\alpha\ (\alpha\ \alpha)\ \lambda\alpha\ (\alpha\ \alpha)$ has type $S$. The only way to reduce this proof is to reduce it to itself and hence it is not normalizable.

An instance of this rewrite rule is skolemized naive set theory. In naive set set theory we have the following axiom scheme

$$\forall x_1\ \ldots\ \forall x_n\ \exists y\ \forall z\ (z \in y \Leftrightarrow P)$$

for any propositional expression $P$.

Skolemizing this scheme, we introduce for each proposition $P$ a symbol $f_{x_1,\ldots,x_n,z,P}$ and an axiom

$$\forall x_1 \ \ldots \ \forall x_n \ \forall z \ (z \in f_{x_1,\ldots,x_n,z,P}(x_1,\ldots,x_n) \Leftrightarrow P)$$

This axiom can be turned into the rewrite rule

$$z \in f_{x_1,\ldots,x_n,z,P}(x_1,\ldots,x_n) \to P$$

In particular we have a rewrite rule

$$z \in f_{z,(z\in z)\Rightarrow\bot} \to (z \in z) \Rightarrow \bot$$

and hence writing $R$ for the proposition $f_{z,(z\in z)\Rightarrow\bot} \in f_{z,(z\in z)\Rightarrow\bot}$ and $S$ for the proposition $\bot$ we have

$$R \to R \Rightarrow S.$$

We thus reconstructed Russell's counter-example to consistency and cut elimination for naive set theory.

*Example:* (Crabbé's counter-example)

Even if Zermelo's set theory is considered coherent, it is well-known that cut elimination is problematic and does generally not hold. The proof of non normalization is called Crabbé's counter-example (see [10, 4] for details). Again, it is here illustrated by the fact that the straightforward encoding of set theory as a deduction modulo necessitates a non-terminating rewrite system.

Consider the following rewriting system

$$C \to E \wedge (C \Rightarrow D)$$

Modulo this rewriting system, the proof

$$\lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha) \ (\beta, \lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha))$$

is a proof of $D$ in the context $E$. The only way to reduce this proof is to reduce it to

$$(\mathit{snd}(\beta, \lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha)) \ (\beta, \lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha)))$$

and then to itself

$$(\lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha) \ (\beta, \lambda\alpha \ (\mathit{snd}(\alpha) \ \alpha)))$$

Hence it is not normalizable.

An instance of this example is skolemized set theory. In set theory we have an axiom scheme

$$\forall x_1 \ \ldots \ \forall x_n \ \forall w \ \exists y \ \forall z \ (z \in y \Leftrightarrow (z \in w \wedge P))$$

skolemizing this scheme, we introduce for each proposition $P$ a symbol $f_{x_1,\ldots,x_n,z,P}$ and an axiom

$$\forall x_1 \ \ldots \ \forall x_n \ \forall z \ (z \in f_{x_1,\ldots,x_n,z,P}(x_1,\ldots,x_n,w) \Leftrightarrow (z \in w \wedge P))$$

This axiom can be turned into the rewrite rule

$$z \in f_{x_1,\ldots,x_n,z,P}(x_1,\ldots,x_n,w) \to z \in w \wedge P$$

In particular we have a rewrite rule

$$z \in f_{z,(z\in z)\Rightarrow\bot}(w) \to z \in w \wedge (z \in z \Rightarrow \bot)$$

and hence writing $C$ for the proposition $f_{z,(z\in z)\Rightarrow\bot}(w) \in f_{z,(z\in z)\Rightarrow\bot}(w)$, $D$ for the proposition $\bot$ and $E$ for the proposition $f_{z,(z\in z)\Rightarrow\bot}(w) \in w$ we have

$$C \to E \wedge (C \Rightarrow D)$$

### 2.4 The conjecture

In these examples the rewriting system itself is not terminating, as $R$ (resp. $C$) reduces to a proposition where it occurs. We conjecture that this non termination is responsible for the non termination of reduction of proofs.

**Conjecture 2.1** *If $\mathcal{R}$ is a confluent and normalizing rewrite system, then proof reduction modulo $\mathcal{R}$ is normalizing.*

An obvious consequence is that deduction modulo $\mathcal{R}$ is coherent, by theorem 2.1.

## 3 Proof normalization for the intuitionistic natural deduction

Now we want to prove some particular cases of the conjecture. First that proofs normalize for the definition of higher-order logic given above. Then, that proofs normalize for all rewrite systems reducing terms to terms and atomic propositions to *quantifier free* propositions (as in the simplification example above). At last, that proofs normalize for all rewrite systems reducing terms to terms and atomic propositions to *positive* propositions, i.e. one not containing the symbol $\Rightarrow$.

$$(\lambda\alpha\ \pi_1\ \pi_2) \rhd [\pi_2/\alpha]\pi_1$$
$$\mathit{fst}(\pi_1, \pi_2) \rhd \pi_1$$
$$\mathit{snd}(\pi_1, \pi_2) \rhd \pi_2$$
$$(\delta\ i(\pi_1), \alpha\pi_2, \beta\pi_3) \rhd [\pi_1/\alpha]\pi_2$$
$$(\delta\ j(\pi_1), \alpha\pi_2, \beta\pi_3) \rhd [\pi_1/\beta]\pi_3$$
$$(\lambda x\ \pi\ t) \rhd [t/x]\pi$$
$$(\mathit{exelim}\ (t, \pi_1)\ \alpha x\pi_2) \rhd [t/x, \pi_1/\alpha]\pi_2$$

$$(\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3) \rhd \pi_2$$
$$(\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3) \rhd \pi_3$$
$$(\mathit{exelim}\ \pi_1\ x\alpha\pi_2) \rhd \pi_2$$

Figure 5: Ultrareduction rules

### 3.1 Ultrareduction

In order to allow the lift of the normalization theorem from natural deduction to natural deduction with commutative cuts and to the sequent calculus, we need to generalize slightly the result and prove strong normalization for *ultrareductions*.

Ultrareductions are inspired by Girard's thesis [8], and are obtained by adding to the rules above three extra rules (figure 5). Obviously strong normalization for ultrareduction implies that of ordinary reduction.

We write $\mathcal{SN}$ for the set of strongly normalizing proofs.

### 3.2 Reducibility

The basic tools used hereafter are the ones of reducibility proofs, whose main concepts are due to Tait [17] and Girard [8, 9]. In particular, since we want to treat the case of higher-order logic, we need some form of reducibility candidates. We here take a definition similar to [9], but other ones like Tait's saturated sets would also apply [17].

**Definition 3.1** *(Neutral proof)*
*A proof is said to be* neutral *if its last rule is an axiom or an elimination, but not an introduction.*

**Definition 3.2** *(Reducibility candidate)*
*A set $R$ of proofs is a* reducibility candidate *if*

- *if $\pi \in R$, then $\pi$ is strongly normalizable,*
- *if $\pi \in R$ and $\pi \rhd \pi'$ then $\pi' \in R$,*
- *if $\pi$ is neutral and if for every $\pi'$ such that $\pi \rhd^1 \pi'$, $\pi' \in R$ then $\pi \in R$.*

*Let $\mathcal{C}$ be the set of all reducibility candidates.*

Mostly, we follow the main scheme of reducibility proofs. That is we try to construct for every proposition $A$ a set of proofs $\mathcal{R}_A$ such that:

- All elements of $\mathcal{R}_A$ are strongly normalizing.
- If $\Gamma \vdash_{\equiv} t : A$ holds, then $t \in \mathcal{R}_A$.

The first condition is ensured by verifying that all $\mathcal{R}_A$ are reducibility candidates. The second one is proved by induction over the derivation of $\Gamma \vdash_{\equiv} t : A$ using closure conditions due to the definition of $\mathcal{R}_A$. Typically:

- If $\pi \in \mathcal{R}_{A \Rightarrow B}$ and $\pi$ reduces to a proof of the form $\lambda\alpha\pi_1$, then for each proof $\pi'$ of $\mathcal{R}_A$, $[\pi'/\alpha]\pi_1$ is an element of $\mathcal{R}_B$.
- If $\pi \in \mathcal{R}_{A \wedge B}$ and $\pi$ reduces to a proof of the form $(\pi_1, \pi_2)$, then $\pi_1$ is an element of $\mathcal{R}_A$ and $\pi_2$ is an element of $\mathcal{R}_B$.
- If $\pi \in \mathcal{R}_{A \vee B}$ and $\pi$ reduces to a proof of the form $i(\pi_1)$ (resp. $j(\pi_2)$) then $\pi_1$ is an element of $\mathcal{R}_A$ (resp. $\pi_2$ is an element of $\mathcal{R}_B$).
- If $\pi \in \mathcal{R}_{\forall x\ A}$ and $\pi$ reduces to a proof of the form $\lambda x\ \pi_1$ then for each term $t$ of the same sort than $x$, $[t/x]\pi_1$ is an element of $\mathcal{R}_{[t/x]A}$.
- If $\pi \in \mathcal{R}_{\exists x\ A}$ and $\pi$ reduces to a proof of the form $(t, \pi_1)$ then $[t/x]\pi_1$ is an element of $\mathcal{R}_{[t/x]A}$.

If we read the conditions above as a partial definition of the family of sets $\mathcal{R}_A$, we understand that the crucial step will be choosing the right sets for $\mathcal{R}_A$ in the case where $A$ is atomic. In first-order logic, we usually take $\mathcal{SN}$ for the set $\mathcal{R}_A$ when $A$ is atomic, but here this is not possible. Indeed, an atomic proposition $A$ can be reduced to a non atomic one $B \Rightarrow C$ and thus a proof $\pi_1$ of $A$ may be applied to

a proof $\pi_2$ of $B$ to form a proof $(\pi_1\ \pi_2)$ of $C$. The strong normalization of $(\pi_1\ \pi_2)$ cannot be deduced from that of $\pi_1$ and that of $\pi_2$ because the root may be a redex. Hence, as $A$ rewrites to $B \Rightarrow C$ we have to take the same conditions on $\mathcal{R}_A$ than on $\mathcal{R}_{B\Rightarrow C}$. A solution is to take $\mathcal{R}_A = \mathcal{R}_{B\Rightarrow C}$.

More generally we require that $\mathcal{R}_A = \mathcal{R}_B$ when $A \equiv B$.

To define the family $\mathcal{R}_A$, it is enough to define for every predicate symbol $P$ the sets $\mathcal{R}_{P(t_1,\ldots,t_n)}$, or equivalently to give, for each $n$-ary predicate symbol $P$, a function $\hat{P}$ that maps $n$-uples of terms to some well-chosen reducibility candidate.

It is well-know that a reducibility proof essentially boils down to the construction of a particular syntactical model. This comparison is particularly striking here since, in first-order logic, to define a model, we also need to provide, for each predicate symbol $P$ a function $\hat{P}$ that maps every $n$-tuple of terms to a truth value.

We can pursue this comparison. If two terms $t_1$ and $t'_1$ are congruent then the sets $\hat{P}(t_1,\ldots,t_n)$ and $\hat{P}(t'_1,\ldots,t_n)$ must be identical. The function $\hat{P}$ is then better defined as a function from an abstract object (for instance, the class of $t_1$ and $t'_1$) that $t_1$ and $t'_1$ denote.

Then the condition that two congruent propositions must have the same denotation can be expressed as the fact that the congruence is valid in the model.

### 3.3 Pre-model

Formalizing the discussion above, we end-up with the following notion.

**Definition 3.3** *(Pre-model)*
*Let $\mathcal{L}$ be a (many sorted) first-order language. A* pre-model *for $\mathcal{L}$ is given by:*

- *for each sort $T$, a set $\mathcal{M}_T$,*
- *for each function symbol $f$ (of rank $(T_1,\ldots,T_n,U)$), a function $\hat{f}$ element of the set $\mathcal{M}_U^{\mathcal{M}_{T_1}\times\ldots\times\mathcal{M}_{T_n}}$,*
- *for each predicate symbol $P$ (of rank $(T_1,\ldots,T_n)$), a function $\hat{P}$ element of the set $\mathcal{C}^{\mathcal{M}_{T_1}\times\ldots\times\mathcal{M}_{T_n}}$.*

**Definition 3.4** *Let $t$ be a term and $\varphi$ an assignment mapping all the free variables of $t$ of sort $T$ to elements of $\mathcal{M}_T$. We define the object $|t|_\varphi$ by induction over the structure of $t$.*

- $|x|_\varphi = \varphi(x)$,

- $|f(t_1,\ldots,t_n)|_\varphi = \hat{f}(|t_1|_\varphi,\ldots,|t_n|_\varphi)$.

**Definition 3.5** *Let $A$ be a proposition and $\varphi$ an assignment mapping all the free variables of $A$ of sort $T$ to elements of $\mathcal{M}_T$. We define the set $|A|_\varphi$ of proofs by induction over the structure of $A$.*

- *A proof $\pi$ is an element of $|P(t_1,\ldots,t_n)|_\varphi$ if it is in $\hat{P}(|t_1|_\varphi,\ldots,|t_n|_\varphi)$.*
- *A proof $\pi$ is element of $|A \Rightarrow B|_\varphi$ if it is strongly normalizable and when $\pi$ reduces to a proof of the form $\lambda\alpha\pi_1$ then for every $\pi'$ in $|A|_\varphi$, $[\pi'/\alpha]\pi_1$ is an element of $|B|_\varphi$.*
- *A proof $\pi$ is an element of $|A \wedge B|_\varphi$ if it is strongly normalizable and when $\pi$ reduces to a proof of the form $(\pi_1,\pi_2)$ then $\pi_1$ and $\pi_2$ are elements of $|A|_\varphi$ and $|B|_\varphi$.*
- *A proof $\pi$ is an element of $|A \vee B|_\varphi$ if it is strongly normalizable and when $\pi$ reduces to a proof of the form $i(\pi_1)$ (resp. $j(\pi_2)$) then $\pi_1$ (resp. $\pi_2$) is an element of $|A|_\varphi$ (resp. $|B|_\varphi$).*
- *A proof $\pi$ is an element of $|\bot|_\varphi$ if it is strongly normalizable.*
- *A proof $\pi$ is an element of $|\forall x\ A|_\varphi$ if it is strongly normalizable and when $\pi$ reduces to a proof of the form $\lambda x\pi_1$ then for every term $t$ of sort $T$ (where $T$ is the sort of $x$) and every element $E$ of $\mathcal{M}_T$, $[t/x]\pi_1$ is an element of $|A|_{\varphi+(x,E)}$.*
- *A proof $\pi$ is an element of $|\exists x\ A|_\varphi$ if it is strongly normalizable and when $\pi$ reduces to a proof of the form $(t,\pi_1)$ then for every element $E$ of $\mathcal{M}_T$ (where $T$ is the sort of $t$) $\pi_1$ is an element of $|A|_{\varphi+(x,E)}$.*

**Definition 3.6** *A pre-model is a pre-model of $\equiv$ if when $A \equiv B$ then for every assignment $\varphi$, $|A|_\varphi = |B|_\varphi$.*

**Proposition 3.1** *For evey proposition $A$ and assignment $\varphi$, $|A|_\varphi$ is a reducibility candidate*

*Proof.* By induction over the structure of $A$.

If $A$ is an atomic proposition, $|A|_\varphi$ is a reducibility candidate by definition.

If $A$ is a composed proposition, then, by definition, $|A|_\varphi$ contains only normalizable proofs. It is routine to prove that it is closed by reduction.

Now, we assume that $\pi$ is a neutral proof and for every $\pi'$ such that $\pi \triangleright^1 \pi'$, $\pi' \in |A|_\varphi$ and we want to prove that $\pi$ is in $|A|_\varphi$. Following the definition of $|A|_\varphi$, we first prove that $\pi$ is strongly normalizable and then that if it reduces to an introduction, the subproofs belong to the appropriate sets.

We first prove that $\pi$ is strongly normalizable. Let $\pi = \pi_1, \pi_2, \ldots$ be a reduction sequence issued from $\pi$. If this sequence is empty it is finite. Otherwise we have $\pi \triangleright^1 \pi_2$ and hence $\pi_2$ is an element of $|A|_\varphi$ thus it is strongly normalizable and the reduction sequence is finite.

Then we prove that if $\pi$ reduces to a introduction then the subproofs belong to the appropriate sets. Let $\pi = \pi_1, \pi_2, \ldots \pi_n$ be a reduction sequence issued from $\pi$ and such that $\pi_n$ is an introduction. This sequence cannot be empty because $\pi$ is neutral and hence not an introduction. Thus $\pi \triangleright^1 \pi_2 \triangleright \pi_n$. We have $\pi_2 \in |A|_\varphi$ and thus if $\pi_n$ is an introduction the subproofs belong to the appropriate sets. □

**Proposition 3.2** *(Substitution)*

$$|[t/x]A|_\varphi = |A|_{\varphi+(x,|t|_\varphi)}$$

*Proof.* By induction on the structure of $A$. □

## 3.4 The normalization theorem

In this section we prove that if a system has a pre-model then proofs modulo this system normalize.

**Theorem 3.1** *Let $\equiv$ be a congruence, $\mathcal{M}$ be a pre-model of $\equiv$, $A$ be a proposition, $\pi$ be a proof of $A$ modulo $\equiv$, $\theta$ be a substitution mapping the free variables of sort $T$ of $A$ to terms of sort $T$, $\varphi$ be an assignment mapping free variables of $A$ to elements of $\mathcal{M}_T$ and $\sigma$ a substitution mapping proof variables of propositions $B$ to elements of $|B|_\varphi$. Then $\sigma\theta\pi$ is an element of $|A|_\varphi$.*

*Proof.* By induction over the structure of $\pi$; we detail the various cases below.

### 3.4.1 Axiom

If $\pi$ is a variable $\alpha$, then $\sigma\theta\pi = \sigma\alpha$. If $\alpha$ is bound by $\sigma$ then $\sigma\theta\pi$ is an element of $|A|_\varphi$ by definition. If $\alpha$ is not bound by $\sigma$ then $\sigma\theta\pi = \alpha$ and thus it is in $|A|_\varphi$.

### 3.4.2 ⇒-intro

The proof $\pi$ has the form $\lambda\alpha\rho$ where $\alpha$ is a proof variable of some proposition $B$ and $\rho$ a proof of some proposition $C$. We have $\sigma\theta\pi = \lambda\alpha\sigma\theta\rho$, consider a reduction sequence issued from this proof. This sequence can only reduce the proof $\sigma\theta\rho$. By induction hypothesis, the proof $\sigma\theta\rho$ is an element of $|C|_\varphi$, thus the reduction sequence is finite.

Furthermore, every reduct of $\sigma\theta\pi$ is of the form $\lambda\alpha\rho'$ where $\rho'$ is a reduct of $\sigma\theta\rho$. Let then $\tau$ be any proof of $|B|_\varphi$, the proof $[\tau/\alpha]\rho'$ can be obtained by reduction from $([\tau/\alpha]\circ\sigma)\theta\rho$. By induction hypothesis, the proof $([\tau/\alpha]\circ\sigma)\theta\rho$ is an element of $|C|_\varphi$. Hence, as $|C|_\varphi$ is a reducibility candidate, the proof $[\tau/\alpha]\rho'$ is an element of $|C|_\varphi$.

Hence, the proof $\sigma\theta\pi$ is an element of $|A|_\varphi$.

### 3.4.3 ∧-intro

The proof $\pi$ has the form $(\rho_1, \rho_2)$ where $\rho_1$ is a proof of some proposition $B$ and $\rho_2$ a proof of some proposition $C$. We have $\sigma\theta\pi = (\sigma\theta\rho_1, \sigma\theta\rho_2)$. Consider a reduction sequence issued from this proof. This sequence can only reduce the proofs $\sigma\theta\rho_1$ and $\sigma\theta\rho_2$. By induction hypothesis these proofs are in $|B|_\varphi$ and $|C|_\varphi$. Thus the reduction sequence is finite.

Furthermore, any reduct of $\sigma\theta\pi$ is of the form $(\rho'_1, \rho'_2)$ where $\rho'_1$ is a reduct of $\sigma\theta\rho_1$ and $\rho'_2$ one of $\rho_2$. These proofs are in $|B|_\varphi$ and $|C|_\varphi$ because these sets are candidates.

Hence, the proof $\sigma\theta\pi$ is in $|A|_\varphi$.

### 3.4.4 ∨-intro

The proof $\pi$ has the form $i(\rho)$ (resp. $j(\rho)$) and $\rho$ is a proof of some proposition $B$. We have $\sigma\theta\pi = i(\sigma\theta\rho)$ (resp. $j(\sigma\theta\rho)$). Consider a reduction sequence issued from this proof. This sequence can only reduce the proofs $\sigma\theta\rho$. By induction hypothesis this proof is an element of $|B|_\varphi$. Thus the reduction sequence is finite.

Furthermore all reducts of $\sigma\theta\pi$ are of the form $i(\rho')$ (resp. $j(\rho')$) where $\rho'$ is a reduct of $\sigma\theta\rho$. This proof is an element of $|B|_\varphi$ since this set is a candidate.

Hence, the proof $\sigma\theta\pi$ is an element of $|A|_\varphi$.

### 3.4.5 ∀-intro

The proof $\pi$ has the form $\lambda x\rho$ where $\rho$ is a proof of some proposition $B$. We have $\sigma\theta\pi = \lambda x\sigma\theta\rho$.

Consider a reduction sequence issued from the proof $\sigma\theta\pi = \lambda x \sigma\theta\rho$. This sequence can only reduce the proof $\sigma\theta\rho$. Let $E$ an element of $\mathcal{M}_T$ (where $T$ is the sort of $x$). By induction hypothesis, the proof $\sigma\theta\rho$ is an element of $|B|_{\varphi+(x,E)}$, thus the reduction sequence is finite.

All reducts of $\sigma\theta\pi$ are of the form $\lambda x \rho'$ where $\rho'$ is a reduct of $\sigma\theta\rho$. The proof $[t/x]\rho'$ is obtained by reducing the proof $[t/x]\sigma\theta\rho$. By induction hypothesis again, the proof $[t/x]\sigma\theta\rho$ is an element of $|B|_{\varphi+(x,E)}$. Thus $\lambda x\ \sigma\theta\rho$ is an element of $|A|_\varphi$.

Hence, the proof $\sigma\theta\pi$ is an element of $|A|_\varphi$.

### 3.4.6 $\exists$-intro

The proof $\pi$ has the form $(t, \rho)$ and $\sigma\theta\pi = (\sigma\theta t, \sigma\theta\rho)$. Consider a reduction sequence issued from this proof. This sequence can only reduce the proof $\sigma\theta\rho$. By induction hypothesis this proof is in $|[t/x]A|_\varphi$. Thus the reduction sequence is finite.

Furthermore, any reduct of $\sigma\theta\pi$ is of the form $(\sigma\theta t, \rho')$ where $\rho'$ is a reduct of $\sigma\theta\rho$. This proof is an element of $|[t/x]A|_\varphi$ because $|[t/x]A|_\varphi$ is a candidate.

Hence, the proof $\sigma\theta\pi$ is an element of $|A|_\varphi$.

### 3.4.7 $\Rightarrow$-elim

The proof $\pi$ has the form $(\rho_1\ \rho_2)$ and $\rho_1$ is a proof of some proposition $B \Rightarrow A$ and $\rho_2$ a proof of the proposition $B$. We have $\sigma\theta\pi = (\sigma\theta\rho_1\ \sigma\theta\rho_2)$. By induction hypothesis $\sigma\theta\rho_1$ and $\sigma\theta\rho_2$ are in the sets $|B \Rightarrow A|_\varphi$ and $|B|_\varphi$. Hence these proofs are strongly normalizable. Let $n$ be the maximum length of a reduction sequence issued from $\sigma\theta\rho_1$ and $n'$ the maximum length of a reduction sequence issued from $\sigma\theta\rho_2$. We prove by induction on $n + n'$ that $(\sigma\theta\rho_1\ \sigma\theta\rho_2)$ is in the set $|A|_\varphi$. As this proof is neutral we only need to prove that every of its one step reducts is in $|A|_\varphi$. If the reduction takes place in $\sigma\theta\rho_1$ or in $\sigma\theta\rho_2$ then we apply the induction hypothesis. Otherwise $\sigma\theta\rho_1$ has the form $\lambda\alpha\ \rho'$ and the reduct is $[\sigma\theta\rho_2/\alpha]\rho'$. By the definition of $|B \Rightarrow A|_\varphi$ this proof is in $|A|_\varphi$.

Hence, the proof $\sigma\theta\pi$ is an element of $|A|_\varphi$.

### 3.4.8 $\wedge$-elim

We only detail the case of left elimination. The proof $\pi$ has the form $fst(\rho)$ where $\rho$ is a proof of some proposition $A \wedge B$. We have $\sigma\theta\pi = fst(\sigma\theta\rho)$. By induction hypothesis the proof $\sigma\theta\rho$ is in $|A \wedge B|_\varphi$. Hence, it is strongly normalizable. Let $n$ be the maximum length of a reduction sequence issued from this proof. We prove by induction on $n$ that $fst(\sigma\theta\rho)$ is in the set $|A|_\varphi$. Since this proof is neutral we only

need to prove that every of its one step reducts is in $|B|_\varphi$. If the reduction takes place in $\sigma\theta\rho$ then we apply the induction hypothesis. Otherwise $\sigma\theta\rho$ has the form $(\rho'_1, \rho'_2)$ and the reduct is $\rho'_1$. By the definition of $|A \wedge B|_\varphi$ this proof is in $|A|_\varphi$.

Hence, the proof $\sigma\theta\pi$ is an element of $|A|_\varphi$.

### 3.4.9 $\vee$-elim

The proof $\pi$ has the form $(\delta\ \rho_1\ \alpha\rho_2\ \beta\rho_3)$ where $\rho_1$ is a proof of some proposition $B \vee C$ and $\rho_2$ and $\rho_3$ are proofs of $A$. We have $\sigma\theta\pi = (\delta\ \sigma\theta\rho_1\ \alpha\sigma\theta\rho_2\ \beta\sigma\theta\rho_3)$. By induction hypothesis, the proof $\sigma\theta\rho_1$ is in the set $|B \vee C|_\varphi$, and the proofs $\sigma\theta\rho_2$ and $\sigma\theta\rho_3$ are in the set $|A|_\varphi$. Hence, these proofs are strongly normalizable. Let $n$, $n'$ and $n''$ be the maximum length of reduction sequences issued from these proofs. We prove by induction on $n + n' + n''$ that $(\delta\ \sigma\theta\rho_1\ \sigma\theta\alpha\rho_2\ \beta\sigma\theta\rho_3)$ is in $|A|_\varphi$. Since this proof is neutral we only need to prove that every of its one step reducts is in $|A|_\varphi$. If the reduction takes place in $\sigma\theta\rho_1$, $\sigma\theta\rho_2$ or $\sigma\theta\rho_3$ then we apply the induction hypothesis. Otherwise, if $\sigma\theta\rho_1$ has the form $i(\rho')$ (resp. $j(\rho')$) and the reduct is $[\rho'/\alpha]\sigma\theta\rho_2$ (resp. $[\rho'/\beta]\sigma\theta\rho_3$). By the definition of $|B \vee C|_\varphi$ the proof $\rho'$ is in $|B|_\varphi$ (resp. $|C|_\varphi$). Hence by induction hypothesis $([\rho'/\alpha] \circ \sigma)\theta\rho_2$ (resp. $([\rho'/\beta] \circ \sigma)\theta\rho_3$) is in $|A|_\varphi$.

At last, if the proof is reduced by the extended rules the reduct is $\sigma\theta\rho_2$ or $\sigma\theta\rho_3$ and these proofs are in $|A|_\varphi$.

Hence, the proof $\sigma\theta\pi$ is an element of $|A|_\varphi$.

### 3.4.10 $\perp$-elim

The proof $\pi$ has the form $(\mathit{botelim}\ \rho)$ with $\rho$ being a proof of $\perp$. We have $\sigma\theta\pi = (\mathit{botelim}\ \sigma\theta\rho)$. By induction hypothesis, the proof $\sigma\theta\rho$ is an element of $|\perp|_\varphi$. Hence, it is strongly normalizable. Let $n$ be the maximum length of reduction sequences issued from this proof. We prove by induction on $n$ that $(\mathit{botelim}\ \sigma\theta\rho)$ is in $|A|_\varphi$. Since this proof is neutral, we only need to prove that every of its one step reducts is in $|A|_\varphi$. The reduction can only take place in $\sigma\theta\rho$ and we apply the induction hypothesis.

Hence, the proof $\sigma\theta\pi$ is an element of $|A|_\varphi$.

### 3.4.11 $\forall$-elim

The proof $\pi$ has the form $(\rho\ t)$ where $\rho$ is a proof of some proposition $\forall x\ B$ and $A = [t/x]B$. We have $\sigma\theta\pi = (\sigma\theta\rho\ \theta t)$. By induction hypothesis, the proof $\sigma\theta\rho$ is in $|\forall x\ B|_\varphi$. Hence, it is strongly normalizable. Let $n$ be the maximum length of a

reduction sequence issued from this proof. We prove by induction on $n$ that $(\sigma\theta\rho\ \theta t)$ is in the set $|A|_\varphi$. As this proof is neutral, we only need to prove that every of its one step reducts is in $|A|_\varphi$. If the reduction takes place in $\sigma\theta\rho$ then we apply the induction hypothesis. Otherwise $\sigma\theta\rho$ has the form $\lambda x\ \rho'$ and the reduct is in $[\theta t/x]\rho'$. By the definition of $|\forall x\ A|_\varphi$ this proof is in $|A|_\varphi$.

Hence, the proof $\sigma\theta\pi$ is an element of $|A|_\varphi$.

#### 3.4.12 $\exists$-elim

The proof $\pi$ has the form $(exelim\ \rho_1\ \alpha x\rho_2)$ where $\rho_1$ is a proof of some proposition $\exists x\ B$ and $\rho_2$ is a proof of $A$. We have $\sigma\theta\pi = (exelim\ \sigma\theta\rho_1\ \alpha x\sigma\theta\rho_2)$. By induction hypothesis, the proof $\sigma\theta\rho_1$ is in the set $|\exists x\ B|_\varphi$ and the proof $\sigma\theta\rho_2$ is in the set $|A|_\varphi$. Hence, these proofs are strongly normalizable. Let $n$ and $n'$ be the maximum length of reduction sequences issued from these proofs. We prove by induction on $n + n'$ that $(exelim\ \sigma\theta\rho_1\ x\alpha\sigma\theta\rho_2)$ is in $|A|_\varphi$. As this proof is neutral we only need to prove that every of its one step reducts is in $|A|_\varphi$. If the reduction takes place in $\sigma\theta\rho_1$ or $\sigma\theta\rho_2$ then we apply the induction hypothesis. Otherwise, if $\sigma\theta\rho_1$ has the form $(t, \rho')$ and the reduct is $[\rho'/\alpha][t/x]\sigma\theta\rho_2 = ([\rho'/\alpha] \circ [t/x]\sigma)([t/x] \circ \theta)\rho_2$. By the definition of $|\exists x\ A|_\varphi$, the proof $\rho'$ is in $|B|_\varphi$. By induction hypothesis the proof $([\rho'/\alpha] \circ [t/x]\sigma)([t/x] \circ \theta)\rho_2$ is in $|A|_\varphi$. At last, if the reduction rule is an extended one, then the proof reduces to $\sigma\theta\rho_2$ that is in $|B|_\varphi$.

Hence, the proof $\sigma\theta\pi$ is an element of $|A|_\varphi$. □

**Corollary 3.1** *Every proof of $A$ is in $|A|_\emptyset$ and hence strongly normalizable*

### 3.5 Pre-model construction

Constructing the pre-model for a given theory, is the part of the consistency proof that bears the logical complexity; i.e. it is the part of the proof that cannot be done in the theory itself. The construction for higher-order logic follows essentially the original proof of Girard. The two other ones we present are more typical of deduction modulo.

#### 3.5.1 Higher-order logic

**Proposition 3.3** *Higher-order logic has a pre-model, hence proofs normalize in higher-order logic.*

*Proof.* We construct a pre-model as follows. The essential point is that we anticipate the fact that objects of sort $o$ actually represent propositions, by interpreting them as reducibility candidates. Thus quantification over $o$ becomes impredicative in the model.

$$\begin{aligned}
\mathcal{M}_\iota &= \{0\} \\
\mathcal{M}_o &= \mathcal{C} \\
\mathcal{M}_{T\to U} &= \mathcal{M}_U^{\mathcal{M}_T}
\end{aligned}$$

$$\begin{aligned}
\hat{S} &= a \mapsto (b \mapsto (c \mapsto a(c)(b(c)))) \\
\hat{K} &= a \mapsto (b \mapsto a) \\
\hat{\alpha}(a,b) &= a(b) \\
\hat{\varepsilon}(a) &= a
\end{aligned}$$

$$\begin{aligned}
\hat{\dot{\Rightarrow}}(a,b) &= \{\pi \in \mathcal{SN} | \pi \rhd \lambda\alpha\pi_1 \Rightarrow \forall \pi' \in a.[\pi'/\alpha]\pi_1 \in b\} \\
\hat{\dot{\wedge}}(a,b) &= \{\pi \in \mathcal{SN} | \pi \rhd (\pi_1,\pi_2) \Rightarrow \pi_1 \in a \wedge \pi_2 \in b\} \\
\hat{\dot{\vee}}(a,b) &= \{\pi \in \mathcal{SN} | (\pi \rhd i(\pi_1) \Rightarrow \pi_1 \in a) \wedge (\pi \rhd i(\pi_2) \Rightarrow \pi_2 \in b)\} \\
\hat{\dot{\bot}} &= \mathcal{SN} \\
\hat{\dot{\forall}}_T(a) &= \{\pi \in \mathcal{SN} | \pi \rhd \lambda x\pi_1 \Rightarrow \forall t : T.\forall E \in \mathcal{M}_T.[t/x]\pi_1 \in a(E)\} \\
\hat{\dot{\exists}}_T(a) &= \{\pi \in \mathcal{SN} | \pi \rhd (\pi_1,\pi_2) \Rightarrow \pi_1 \in \mathcal{M}_T \wedge \forall E \in \mathcal{M}_T.\pi_2 \in a(E)\}
\end{aligned}$$

We do not detail the proof that if $A \equiv B$ then $|A|_\varphi = |B|_\varphi$. □

In section 4.2, we study cut elimination for classical sequent calculus. This is done using a, so-called, $\neg\neg$-translation. To this matter, we need the following proposition.

**Proposition 3.4** *The variant of higher-order logic with the following rewrite system has a pre-model*

$$\begin{aligned}
\alpha(\alpha(\alpha(S,x),y),z) &\to \alpha(\alpha(x,z),\alpha(y,z)) \\
\alpha(\alpha(K,x),y) &\to x \\
\varepsilon(\alpha(\alpha(\dot{\Rightarrow},x),y)) &\to \neg\neg\varepsilon(x) \Rightarrow \neg\neg\varepsilon(y) \\
\varepsilon(\alpha(\alpha(\dot{\wedge},x),y)) &\to \neg\neg\varepsilon(x) \wedge \neg\neg\varepsilon(y) \\
\varepsilon(\alpha(\alpha(\dot{\vee},x),y)) &\to \neg\neg\varepsilon(x) \vee \neg\neg\varepsilon(y) \\
\varepsilon(\dot{\bot}) &\to \bot
\end{aligned}$$

$$\begin{array}{rcl}\varepsilon(\alpha(\dot\forall, x)) & \to & \forall y\ \neg\neg\varepsilon(\alpha(x,y)) \\ \varepsilon(\alpha(\dot\exists, x)) & \to & \exists y\ \neg\neg\varepsilon(\alpha(x,y))\end{array}$$

*where $\neg A$ is a notation for $A \Rightarrow \bot$.*

*Proof.* We take the same pre-model as above, but for the interpretation of the symbols $\dot\Rightarrow$, $\dot\wedge$, $\dot\vee$, $\dot\bot$, $\dot\forall$ and $\dot\exists$. We first define the following functions.

$$\begin{array}{rcl}\tilde\Rightarrow(a,b) & = & \{\pi \in \mathcal{SN} | \pi \rhd \lambda\alpha\pi_1 \Rightarrow \forall \pi' \in a.[\pi'/\alpha]\pi_1 \in b\} \\ \tilde\wedge(a,b) & = & \{\pi \in \mathcal{SN} | \pi \rhd (\pi_1,\pi_2) \Rightarrow \pi_1 \in a \wedge \pi_2 \in b\} \\ \tilde\vee(a,b) & = & \{\pi \in \mathcal{SN} | (\pi \rhd i(\pi_1) \Rightarrow \pi_1 \in a) \wedge (\pi \rhd i(\pi_2) \Rightarrow \pi_2 \in b)\} \\ \tilde\bot & = & \mathcal{SN} \\ \tilde\forall_T(a) & = & \{\pi \in \mathcal{SN} | \pi \rhd \lambda x\pi_1 \Rightarrow \forall t : T. \forall E \in \mathcal{M}_T.[t/x]\pi_1 \in a(E)\} \\ \tilde\exists_T(a) & = & \{\pi \in \mathcal{SN} | \pi \rhd (\pi_1,\pi_2) \Rightarrow \pi_1 \in \mathcal{M}_T \wedge \forall E \in \mathcal{M}_T.\pi_2 \in a(E)\} \\ \tilde\neg(a) & = & \tilde\Rightarrow(a, \tilde\bot)\end{array}$$

Then we take

$$\begin{array}{rcl}\hat\Rightarrow(a,b) & = & \tilde\Rightarrow(\tilde\neg(\tilde\neg(a)), \tilde\neg(\tilde\neg(b))) \\ \hat\wedge(a,b) & = & \tilde\wedge(\tilde\neg(\tilde\neg(a)), \tilde\neg(\tilde\neg(b))) \\ \hat\vee(a,b) & = & \tilde\vee(\tilde\neg(\tilde\neg(a)), \tilde\neg(\tilde\neg(b))) \\ \hat\bot & = & \tilde\bot \\ \hat\forall_T(a) & = & \tilde\forall(x \mapsto \tilde\neg(\tilde\neg(a(x)))) \\ \hat\exists_T(a) & = & \tilde\exists(x \mapsto \tilde\neg(\tilde\neg(a(x))))\end{array}$$

□

### 3.5.2 Quantifier free rewrite systems

In the case of higher-order logic, it is the presence of the quantifier $\forall$ on the right hand part of one of the rewrite schemes that is responsible for the impredicativity of the resulting logic. We can give a generic proof of cut elimination for the predicative case:

**Proposition 3.5** *A quantifier free confluent terminating rewrite systems has a pre-model, hence proofs normalize in modulo such a rewrite system.*

*Proof.* For any proposition $A$ (resp. object $t$), let $A\downarrow$ (resp. $t\downarrow$) stand for its normal form. To each normal closed proposition $A$, we associate a set of proofs $\Psi(A)$.

$$\begin{array}{rcl}
\Psi(A) & = & \mathcal{SN} \quad \text{if } A \text{ is atomic} \\
\Psi(A \Rightarrow B) & = & \{\pi \in \mathcal{SN} | \pi \rhd \lambda\alpha.\pi_1 \Rightarrow \forall \pi' \in \Psi(A).[\pi'/\alpha]\pi_1 \in \Psi(B)\} \\
\Psi(A \wedge B) & = & \{\pi \in \mathcal{SN} | \pi \rhd (\pi_1, \pi_2) \Rightarrow \pi_1 \in \Psi(A) \wedge \pi_2 \in \Psi(B)\} \\
\Psi(A \vee B) & = & \{\pi \in \mathcal{SN} | \pi \rhd i(\pi_1) \Rightarrow \pi_1 \in \Psi(A) \wedge \pi \rhd i(\pi_2) \Rightarrow \pi_2 \in \Psi(B)\} \\
\Psi(\bot) & = & \mathcal{SN} \\
\Psi(\forall x\ A) & = & \{\pi \in \mathcal{SN} | \pi \rhd \lambda x^s \pi_1 \Rightarrow \forall t : s.[t/x]\pi_1 \in \Psi(([t/x]A)\downarrow)\} \\
\Psi(\exists x\ A) & = & \{\pi \in \mathcal{SN} | \pi \rhd (t, \pi_1) \Rightarrow \pi_1 \in \Psi([t/x]A\downarrow)\}
\end{array}$$

This definition is well-formed by induction on the pair $(s(A), s'(A))$ where $s(A)$ is the number of quantifiers in $A$ and $s'(A)$ the numbers of connectors.

Then we define a pre-model as follows:

Let $\mathcal{M}_T$ be the set of normal closed terms of sort $T$.

$$\begin{array}{rcl}
\hat{f}(t_1, \ldots, t_n) & = & f(t_1, \ldots, t_n)\downarrow \\
\hat{P}(t_1, \ldots, t_n) & = & \Psi((P(t_1, \ldots, t_n))\downarrow).
\end{array}$$

Again, we leave out the proof that if $A = B$ then $|A|_\varphi = |B|_\varphi$. □

*Remark:* In this normalization proof we use the fact that some sets are reducibility candidates, but we never quantify on all reducibility candidates, reflecting the fact that we here deal with predicative systems.

### 3.5.3 Positive rewrite systems

Finally, for some systems, premodels can be constructed by a fixed point construction.

**Definition 3.7** *A rewrite system is said to be* positive *if it rewrites atomic propositions to propositions containing only positive occurrences of atomic propositions.*

Let $\mathcal{R}$ be a confluent and normalizing rewrite system.

**Definition 3.8** *A pre-model is said to be* syntactical *if*

- *$\mathcal{M}_T$ be the set of normal closed terms of sort $T$,*
- *If $f$ is a function symbol, $\hat{f}$ is the function that maps $t_1, \ldots, t_n$ to the normal form of $f(t_1, \ldots, t_n)$.*

A syntactical pre-model is defined solely by the interpretation of predicate variables.

**Definition 3.9** *Let $\mathcal{M}$ and $\mathcal{M}'$ be two syntactical premodels. We write $\hat{P}$ for the denotation of $P$ in $\mathcal{M}$ and $\overline{P}$ for the denotation of $P$ in $\mathcal{M}'$*

*We say that $\mathcal{M} < \mathcal{M}'$ if and only if for any predicate symbol $P$ and closed terms $t_1, \ldots, t_n$ we have*

$$\hat{P}(t_1, \ldots, t_n) \subset \overline{P}(t_1, \ldots, t_n)$$

The set of syntactical premodels is a complete lattice for the order $<$.

**Definition 3.10** *The functional $\mathcal{F}$ maps syntactical premodels to syntactical premodels. Let $\mathcal{M}$ be a such a premodel, we define the premodel $\mathcal{F}(\mathcal{M})$ by*

$$\mathcal{F}(\mathcal{M})(P)(t_1, \ldots, t_n) = |P(t_1, \ldots, t_n) \downarrow|_{\mathcal{M},\emptyset}$$

**Proposition 3.6** *If the rewrite system is positive then the functional $\mathcal{F}$ is monotone.*

**Proposition 3.7** *If the rewrite system $\mathcal{R}$ is such that the functional $\mathcal{F}$ is monotone, then it has a pre-model, hence proofs normalize modulo such a rewrite system.*

*It is in particular the case of positive systems.*

*Proof.* Assume the functional $\mathcal{F}$ is monotone. Since the set of syntactical premodels is a complete lattice, the function $\mathcal{F}$ has a fixpoint. This fixpoint is can easily proved to be a pre-model of the rewrite system. □

## 3.6 Normalization with commutative cuts

In order to lift the cut elimination theorem from natural deduction to sequent calculus a cut elimination theorem, taking into account the so-called *commutative cuts*.

A usual cut is an introduction rule immediately followed by an elimination rule on the same symbol. A commutative cut is a introduction followed by an elimination, but separated by a sequence of eliminations of symbols $\vee$, $\bot$ or $\exists$. When a proof contains a commutative cut, it is possible to permute the eliminations until the introduction and the elimination form a usual cut that then can be reduced.

For instance if $\pi_1$ is a proof of $A \vee B$, $\pi_2$ a proof of $C \Rightarrow A \Rightarrow D$, and $\pi_3$ a proof of $C \Rightarrow B \Rightarrow D$.

We can form the proof of $C \Rightarrow D$

$$(\delta\ \pi_1\ \alpha(\lambda\gamma\ (\pi_2\ \gamma\ \alpha))\ \beta(\lambda\gamma\ (\pi_3\ \gamma\ \beta)))$$

Now if $\pi_4$ is a proof of $C$ we can form the proof of $D$

$$((\delta\ \pi_1\ \alpha(\lambda\gamma\ (\pi_2\ \gamma\ \alpha))\ \beta(\lambda\gamma\ (\pi_3\ \gamma\ \beta)))\ \pi_4)$$

But, of course, the proposition $D$ has a much simpler proof

$$(\delta\ \pi_1\ \alpha(\pi_2\ \pi_4\ \alpha)\ \beta(\pi_3\ \pi_4\ \beta))$$

This proof can be obtained by permuting the elimination of the disjunction and that of the implication.

$$(\delta\ \pi_1\ \alpha(\lambda\gamma\ (\pi_2\ \gamma\ \alpha)\ \pi_4)\ \beta(\lambda\gamma\ (\pi_3\ \gamma\ \beta)\ \pi_4))$$

and then reducing the usual redexes.

The reduction rules with commutative cuts are given in figure 6.

We want to prove that each proof has a normal form for this rewrite system.

**Definition 3.11** *(Elimination context)*
*An elimination context is a proof of the form*

$$E := () \mid (E\ \pi) \mid \mathit{fst}(E) \mid \mathit{snd}(E) \mid (\delta\ E\ \pi_1\ \pi_2) \mid (\mathit{botelim}\ E) \mid (E\ t) \mid (\mathit{exelim}\ E\ x\alpha\pi)$$

*where* $()$ *is a distinguished variable.*
*We write* $E(\pi)$ *for the proof* $[\pi/()]E$.

**Definition 3.12** *(Simple proof)*
*A simple proof is a proof of the form*

$$S := \alpha \mid (S\ \pi) \mid \mathit{fst}(S) \mid \mathit{snd}(S) \mid (S\ t)$$

**Proposition 3.8** *Every* $\rightarrow$*-normal proof* $\pi$ *that is either an introduction, a simple proof or has the form* $E(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E(\mathit{exelim}\ S\ x\alpha\pi_1)$ *or* $E(\mathit{botelim}\ S)$.

*Proof.* By induction over proof structure.

- If $\pi$ is a variable then it is simple.
- If $\pi$ is an introduction then it is an introduction.

$$(\lambda\alpha\ \pi_1\ \pi_2) \hookrightarrow [\pi_2/\alpha]\pi_1$$
$$\mathit{fst}(\pi_1, \pi_2) \hookrightarrow \pi_1$$
$$\mathit{snd}(\pi_1, \pi_2) \hookrightarrow \pi_2$$
$$(\delta\ i(\pi_1), \alpha\pi_2, \beta\pi_3) \hookrightarrow [\pi_1/\alpha]\pi_2$$
$$(\delta\ j(\pi_1), \alpha\pi_2, \beta\pi_3) \hookrightarrow [\pi_1/\beta]\pi_3$$
$$(\lambda x\ \pi\ t) \hookrightarrow [t/x]\pi$$
$$(\mathit{exelim}\ (t, \pi_1)\ \alpha x\pi_2) \hookrightarrow [t/x, \pi_1/\alpha]\pi_2$$

$$((\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)\ \pi) \hookrightarrow (\delta\ \pi_1\ \alpha(\pi_2\ \pi)\ \beta(\pi_3\ \pi))$$
$$\mathit{fst}(\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3) \hookrightarrow (\delta\ \pi_1\ \alpha\mathit{fst}(\pi_2)\ \beta\mathit{fst}(\pi_3))$$
$$\mathit{snd}(\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3) \hookrightarrow (\delta\ (\pi_1\ \alpha\mathit{snd}(\pi_2)\ \beta\mathit{snd}(\pi_3)))$$
$$(\delta\ (\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)\ \pi_1'\ \pi_2') \hookrightarrow (\delta\ \pi_1\ \alpha(\delta\ \pi_2\ \alpha'\pi_1'\ \beta'\pi_2')\ \beta(\delta\ \pi_3\ \alpha'\pi_1'\ \beta'\pi_2'))$$
$$(\mathit{botelim}\ (\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)) \hookrightarrow (\delta\ \pi_1\ \alpha(\mathit{botelim}\ \pi_2)\ \beta(\mathit{botelim}\ \pi_3))$$
$$((\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)\ t) \hookrightarrow ((\delta\ \pi_1\ \alpha(\pi_2\ t)\ \beta(\pi_3\ t)))$$
$$(\mathit{exelim}\ (\delta\ \pi_1\ \alpha\pi_2\ \beta\pi_3)\ x\alpha'\pi) \hookrightarrow (\delta\ \pi_1\ \alpha(\mathit{exelim}\ \pi_2\ x\alpha'\pi)\ \beta(\mathit{exelim}\ \pi_3\ x\gamma\pi))$$

$$((\mathit{exelim}\ \pi_1\ x\alpha\pi_2)\ \pi) \hookrightarrow (\mathit{exelim}\ \pi_1\ \alpha(\pi_2\ \pi))$$
$$\mathit{fst}(\mathit{exelim}\ \pi_1\ x\alpha\pi_2) \hookrightarrow (\mathit{exelim}\ \pi_1\ x\alpha\mathit{fst}(\pi_2))$$
$$\mathit{snd}(\mathit{exelim}\ \pi_1\ x\alpha\pi_2) \hookrightarrow (\mathit{exelim}\ \pi_1\ x\alpha\mathit{snd}(\pi_2))$$
$$(\delta\ (\mathit{exelim}\ \pi_1\ x\alpha\pi_2)\ \alpha'\pi_1'\ \beta'\pi_2') \hookrightarrow (\mathit{exelim}\ \pi_1\ x\alpha(\delta\ \pi_2\ \alpha'\pi_1'\ \beta'\pi_2'))$$
$$(\mathit{botelim}\ (\mathit{exelim}\ \pi_1\ x\alpha\pi_2)) \hookrightarrow (\mathit{exelim}\ \pi_1\ x\alpha(\mathit{botelim}\ \pi_2))$$
$$((\mathit{exelim}\ \pi_1\ x\alpha\pi_2)\ t) \hookrightarrow (\mathit{exelim}\ \pi_1\ x\alpha(\pi_2\ t))$$
$$(\mathit{exelim}\ (\mathit{exelim}\ \pi_1\ x\alpha\pi_2)\ x\alpha'\pi) \hookrightarrow (\mathit{exelim}\ \pi_1\ x\alpha(\mathit{exelim}\ \pi_2\ x\alpha'\pi))$$

$$((\mathit{botelim}\ \pi_1)\ \pi) \hookrightarrow (\mathit{botelim}\ \pi_1)$$
$$\mathit{fst}(\mathit{botelim}\ \pi_1) \hookrightarrow (\mathit{botelim}\ \pi_1)$$
$$\mathit{snd}(\mathit{botelim}\ \pi_1) \hookrightarrow (\mathit{botelim}\ \pi_1)$$
$$(\delta\ (\mathit{botelim}\ \pi_1)\ \alpha'\pi_1'\ \beta'\pi_2') \hookrightarrow (\mathit{botelim}\ \pi_1)$$
$$(\mathit{botelim}\ (\mathit{botelim}\ \pi_1)) \hookrightarrow (\mathit{botelim}\ \pi_1)$$
$$((\mathit{botelim}\ \pi_1)\ t) \hookrightarrow (\mathit{botelim}\ \pi_1)$$
$$(\mathit{exelim}\ (\mathit{botelim}\ \pi_1)\ x\alpha'\pi) \hookrightarrow (\mathit{botelim}\ \pi_1)$$

Figure 6: Reduction rules with commutative cuts

- If $\pi$ is an elimination, then it has the form $(\rho\ \rho_1)$, $fst(\rho)$, $snd(\rho)$, $(\delta\ \rho\ \alpha\rho_1\ \beta\rho_2)$, $(botelim\ \rho)$, $(\rho\ t)$ or $(exelim\ \rho\ x\alpha\rho_1)$.

  In all these cases, by induction hypothesis the proof $\rho$ is either an introduction, a simple proof or has the form $E(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E(exelim\ S\ x\alpha\pi_1)$ or $E(botelim\ S)$.

  Since $\pi$ is normal, $\rho$ is not an introduction.

  If $\rho$ is simple then $\pi$ is either simple or has the form $(\delta\ S\ \alpha\rho_1\ \beta\rho_2)$, $(botelim\ S)$, $(exelim\ S\ x\alpha\rho_1)$.

  If $\rho$ has the form $E(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E(exelim\ S\ x\alpha\pi_1)$ or $E(botelim\ S)$, then $\pi$ has the form $E'(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E'(exelim\ S\ x\alpha\pi_1)$ or $E'(botelim\ S)$.

□

*Remark:* A simple proof can reduce to simple proofs only.

**Proposition 3.9** *(Weak normalization)*
*Every proof has a $\hookrightarrow$-normal form.*

*Proof.* Let $\pi$ be a proof, we write $n(\pi)$ for the length of the longest $\triangleright$-reduction in $\pi$ and $p(\pi)$ for the size of $\pi$. By induction over the lexicographic ordering $(n(\pi), p(\pi))$, we prove that every proof $\pi$ has a $\hookrightarrow$-normal form.

- If the proof $\pi$ is not $\rightarrow$-normal then it $\rightarrow$-reduces to some proof $\pi'$. We have $n(\pi') < n(\pi)$. By the induction hypothesis $\pi'$ has a $\hookrightarrow$-normal form. Hence $\pi$ has a $\hookrightarrow$-normal form.

- If the proof $\pi$ is $\rightarrow$-normal then it is either an introduction, a simple proof or a proof of the form $E(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E(exelim\ S\ x\alpha\pi_1)$ or $E(botelim\ S)$.

  - If $\pi$ is an introduction, it has the form $\lambda\alpha\pi_1$, $(\pi_1, \pi_2)$, $i(\pi_1)$, $j(\pi_1)$, $\lambda x\pi_1$, $(t, \pi_1)$.
    In all theses cases, we have $n(\pi_i) \leq n(\pi)$ and $p(\pi_i) < p(\pi)$ hence by induction hypothesis the proofs $\pi_i$ have a $\hookrightarrow$-normal forms $\pi_i'$. The proof $\lambda\alpha\pi_1'$, (resp. $(\pi_1', \pi_2')$, $i(\pi_1')$, $j(\pi_1')$, $\lambda x\pi_1'$ or $(t, \pi_1')$) is a $\hookrightarrow$-normal form of $\pi$.
  - If $\pi$ is simple then we prove by structural induction that it has a normal form.
    - If it is a variable then it is its own normal form.

  - If it has the form $(S\ \pi_1)$ then by induction hypothesis $S$ has a normal form $S'$. This proof is simple. We have $n(\pi_1) \leq n(\pi)$ and $p(\pi_1) < p(\pi)$, hence, by induction hypothesis, $\pi_1$ has a normal form $\pi_1'$. The proof $(S'\ \pi_1')$ is a normal form of $\pi$.
  - If $\pi$ has the form $fst(S)$ or $snd(S)$ then by induction hypothesis the proof $S$ has a normal form $S'$. This proof is simple. The proof $fst(S')$ or $snd(S')$ is a normal form of $\pi$.
  - If $\pi$ has the form $(S\ t)$ then by induction hypothesis $S$ has a normal form $S'$. This proof is simple. The proof $(S'\ t)$ is a normal form of $\pi$.

- If $\pi$ has the form $E(\delta\ S\ \alpha\pi_1\ \beta\pi_2)$, $E(exelim\ S\ x\alpha\pi_1)$ or $E(botelim\ S)$, then it $\hookrightarrow$-reduces to $(\delta\ S\ \alpha E(\pi_1)\ \beta E(\pi_2))$, $(exelim\ S\ x\alpha E(\pi_1))$ or $(botelim\ S)$.

  We have $n(S) \leq n(\pi)$ and $p(S) < p(\pi)$ hence, by induction hypothesis the proof $S$ has a normal form $S'$. This proof is simple.

  We have $\pi \rhd^+ E(\pi_i)$, hence $n(E(\pi_i)) < n(\pi)$. Thus, by induction hypothesis the proof $E(\pi_i)$ has a $\hookrightarrow$-normal form $\pi_i'$.

  The proof $(\delta\ S'\ \alpha\pi_1'\ \beta\pi_2')$, $(exelim\ S'\ x\alpha\pi_1')$ or $(botelim\ S')$ is a normal form of $\pi$.

□

# 4 Cut elimination in sequent calculus modulo

## 4.1 Cut elimination for the intuitionistic sequent calculus modulo

Following a usual proof, we show that $\hookrightarrow$-normal proofs in natural deduction can be translated as cut free proofs in sequent calculus.
*Remark:* In natural deduction, the context of the main premise of an elimination is the same as that of the conclusion.

**Proposition 4.1** *If a sequent $\Gamma \vdash_{\equiv} P$ has a normal proof in natural deduction modulo, then it has a cut free proof in sequent calculus modulo.*

*Proof.* By induction on the size of the normal proof of $\Gamma \vdash_{\equiv} P$

- If the last rule is an axiom then the result is obvious.

- If the last rule is an introduction rule, we apply the induction hypothesis, to the subproofs and we use the corresponding right rule.
- If the last rule is an elimination rule, then the proof ends with a sequence of elimination rules on the main premise and we consider the first rule that is not an elimination. This rule cannot be a introduction rule because the proof is normal, thus it is an axiom. We focus on the first rule after this axiom.
  - If this rule is a $\Rightarrow$-elim rule. The proof has the shape

$$\dfrac{\dfrac{\overline{\Gamma \vdash_{\equiv} C}\ \text{axiom} \qquad \dfrac{\pi_1}{\Gamma \vdash_{\equiv} A}}{\Gamma \vdash_{\equiv} B}\ \Rightarrow\text{-elim}}{\dfrac{\pi_2}{\Gamma \vdash_{\equiv} P}}$$

    where $C \equiv A \Rightarrow B$.

    Adding the proposition $B$ in every context of $\pi_2$ we get a proof of $\Gamma B \vdash_{\equiv} P$ under the assumption $\Gamma B \vdash_{\equiv} B$, we can close this proof with an axiom and the proof $\pi_2'$ obtained this way is shorter than the proof we started with.

    We apply the induction hypothesis to $\pi_1$ and $\pi_2'$ yielding cut free proofs $\rho_1$ and $\rho_2$ in sequent calculus of $\Gamma \vdash_{\equiv} A$ and $\Gamma B \vdash_{\equiv} P$.

    The context $\Gamma$ contains the proposition $C$. We build the proof

$$\dfrac{\dfrac{\dfrac{\rho_1}{\Gamma \vdash_{\equiv} A} \quad \dfrac{\rho_2}{\Gamma B \vdash_{\equiv} P}}{\Gamma C \vdash_{\equiv} P}\ \Rightarrow\text{-left}}{\Gamma \vdash_{\equiv} P}\ \text{contraction}$$

  - If this rule is a $\wedge$-elim rule. The proof has the shape

$$\dfrac{\dfrac{\dfrac{\overline{\Gamma \vdash_{\equiv} C}\ \text{axiom}}{\Gamma \vdash_{\equiv} A}\ \wedge\text{-elim}}{\pi}}{\Gamma \vdash_{\equiv} P}$$

    where $C \equiv A \wedge B$.

Adding the propositions $A$ and $B$ in every context of $\pi$ we get a proof of $\Gamma AB \vdash_{\equiv} P$ under the assumption $\Gamma AB \vdash_{\equiv} A$, we can close this proof with an axiom and the proof $\pi'$ obtained this way is shorter than the proof we started with.

We apply the induction hypothesis to $\pi'$ yelding a cut free proof $\rho$ in sequent calculus of $\Gamma AB \vdash_{\equiv} P$

The context $\Gamma$ contains the proposition $C$. We build the proof

$$\dfrac{\dfrac{\dfrac{\rho}{\Gamma AB \vdash_{\equiv} P}}{\Gamma C \vdash_{\equiv} P}\wedge\text{-left}}{\Gamma \vdash_{\equiv} P}\text{contraction}$$

– If this rule is a $\vee$-elim rule. The proof has the shape

$$\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} D}\text{axiom} \qquad \dfrac{\dfrac{\pi_1}{\Gamma A \vdash_{\equiv} C}}{\Gamma \vdash_{\equiv} C} \qquad \dfrac{\pi_2}{\Gamma B \vdash_{\equiv} C}}{\dfrac{\pi_3}{\Gamma \vdash_{\equiv} P}}\vee\text{-elim}$$

where $D \equiv A \vee B$.

As the proof is $\hookrightarrow$-normal and $\pi_3$ contains only eliminations, $\pi_3$ is empty and the proof has the shape

$$\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} D}\text{axiom} \qquad \dfrac{\pi_1}{\Gamma A \vdash_{\equiv} P} \qquad \dfrac{\pi_2}{\Gamma B \vdash_{\equiv} P}}{\Gamma \vdash_{\equiv} P}\vee\text{-elim}$$

We apply the induction hypothesis to $\pi_1$ and $\pi_2$ yelding cut free proofs $\rho_1$ and $\rho_2$ in sequent calculus of $\Gamma A \vdash_{\equiv} P$ and $\Gamma B \vdash_{\equiv} P$.

The context $\Gamma$ contains the proposition $D$. We build the proof

$$\dfrac{\dfrac{\dfrac{\rho_1}{\Gamma A \vdash_{\equiv} P} \quad \dfrac{\rho_2}{\Gamma B \vdash_{\equiv} P}}{\Gamma D \vdash_{\equiv} P}\Rightarrow\text{-left}}{\Gamma \vdash_{\equiv} P}\text{contraction}$$

– If this rule is a $\bot$-elim rule. The proof has the shape

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} B}\,\text{axiom}}{\Gamma \vdash_{\equiv} A}\,\bot\text{-elim}}{\dfrac{\pi}{\Gamma \vdash_{\equiv} P}}$$

where $B \equiv \bot$.
As the proof is $\hookrightarrow$-normal and $\pi$ contains only eliminations, $\pi$ is empty and the proof has the shape

$$\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} B}\,\text{axiom}}{\Gamma \vdash_{\equiv} P}\,\bot\text{-elim}$$

The context $\Gamma$ contains the proposition $B$. We build the proof

$$\dfrac{\dfrac{}{\Gamma B \vdash_{\equiv} P}\,\bot\text{-left}}{\Gamma \vdash_{\equiv} P}\,\text{contraction}$$

– If this rule is a $\forall$-elim rule. The proof has the shape

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash_{\equiv} B}\,\text{axiom}}{\Gamma \vdash_{\equiv} [t/x]A}\,(x, A, t)\forall\text{-elim}}{\dfrac{\pi}{\Gamma \vdash_{\equiv} P}}$$

where $B \equiv \forall x\ A$.
Adding the proposition $[t/x]A$ in every context of $\pi$ we get a proof of $\Gamma[t/x]A \vdash_{\equiv} P$ under the assumption $\Gamma[t/x]A \vdash_{\equiv} [t/x]A$, we can close this proof with an axiom and the proof $\pi'$ obtained this way is shorter than the proof we started with.
We apply the induction hypothesis to $\pi'$ and yelding a cut free proof $\rho$ in sequent calculus of $\Gamma[t/x]A \vdash_{\equiv} P$.
The context $\Gamma$ contains the proposition $B$. We build the proof

$$\dfrac{\dfrac{\dfrac{\rho}{\Gamma[t/x]A \vdash_{\equiv} P}\,(x, A, t)\ \forall\text{-left}}{\Gamma B \vdash_{\equiv} P}\,\text{contraction}}{\Gamma \vdash_{\equiv} P}$$

– If this rule is a $\exists$-elim rule. The proof has the shape

$$\dfrac{\dfrac{\overline{\Gamma \vdash_{\equiv} C}\ \text{axiom} \qquad \dfrac{\pi_1}{\Gamma A \vdash_{\equiv} B}}{\Gamma \vdash_{\equiv} B}\ (x, A)\ \exists\text{-elim}}{\dfrac{\pi_2}{\Gamma \vdash_{\equiv} P}}$$

where $C \equiv \exists x A$.
As the proof is $\hookrightarrow$-normal and $\pi_2$ contains only eliminations, $\pi_2$ is empty and the proof has the shape

$$\dfrac{\overline{\Gamma \vdash_{\equiv} C}\ \text{axiom} \qquad \dfrac{\pi_1}{\Gamma A \vdash_{\equiv} P}}{\Gamma \vdash_{\equiv} P}\ (x, A)\ \exists\text{-elim}$$

We apply the induction hypothesis to $\pi_1$ yelding a cut free proof $\rho_1$ in sequent calculus of $\Gamma A \vdash_{\equiv} P$.
The context $\Gamma$ contains the proposition $C$. We build the proof

$$\dfrac{\dfrac{\dfrac{\rho_1}{\Gamma A \vdash_{\equiv} P}\ (x, A)\ \exists\text{-left}}{\Gamma C \vdash_{\equiv} P}\ \text{contraction}}{\Gamma \vdash_{\equiv} P}$$

□

**Corollary 4.1** *If the congruence $\equiv$ has a pre-model, then the cut rule is redundant in intuitionistic sequent calculus modulo $\equiv$.*

*Proof.* Consider a proof of a sequent $\Gamma \vdash_{\equiv} P$ in sequent calculus, this proof can easily be translated into natural deduction. The $\hookrightarrow$-normal form of this proof can be translated as a cut free proof of $\Gamma \vdash_{\equiv} P$ in sequent calculus. Hence the sequent $\Gamma \vdash_{\equiv} P$ has a cut free proof in sequent calculus. □

## 4.2 Cut elimination for the classical sequent calculus modulo

In this section we prove cut elimination for classical sequent calculus. We use a traditional style reduction to intuitionistic cut elimination through a *$\neg\neg$-translation.*

**Definition 4.1** *Let $A$ be a proposition, the* double negation *of $A$ is the proposition obtained by adding a double negation before each subproposition*

- $A' = \neg\neg A$ *if* $A$ *is atomic,*
- $(A \Rightarrow B)' = \neg\neg(A' \Rightarrow B')$,
- $(A \wedge B)' = \neg\neg(A' \wedge B')$,
- $(A \vee B)' = \neg\neg(A' \vee B')$,
- $\bot' = \neg\neg\bot$,
- $(\forall x\ A)' = \neg\neg(\forall x\ A')$,
- $(\exists x\ A)' = \neg\neg(\exists x\ A')$.

*the* light double negation *of* $A$ *is the proposition obtained by adding a double negation before each subproposition except at the root of the proposition*

- $A'' = A$ *if* $A$ *is atomic,*
- $(A \Rightarrow B)'' = A' \Rightarrow B'$,
- $(A \wedge B)'' = A' \wedge B'$,
- $(A \vee B)'' = A' \vee B'$,
- $\bot'' = \bot$,
- $(\forall x\ A)'' = \forall x\ A'$,
- $(\exists x\ A)'' = \exists x\ A'$.

To a rewrite system $\mathcal{R}$ $A_i \to P_i$ we associate the rewrite system $\mathcal{R}'$ $A_i \to P_i''$.

**Proposition 4.2** *If* $A \to_{\mathcal{R}} B$ *then* $A' \to_{\mathcal{R}'} B'$ *and* $A'' \to_{\mathcal{R}'} B''$.

*Proof.* By induction on the structure of $A$. □

**Corollary 4.2** *If* $A \equiv_{\mathcal{R}} B$ *then* $A' \equiv_{\mathcal{R}'} B'$ *and* $A'' \equiv_{\mathcal{R}'} B''$.

**Proposition 4.3** *If* $\mathcal{R}$ *is terminating and confluent, then so is* $\mathcal{R}'$.

*Proof.* From a reduction sequence in $\mathcal{R}'$ we can build one of the same length in $\mathcal{R}$. Hence all reduction sequences in $\mathcal{R}'$ are finite and $\mathcal{R}'$ is terminating.

By Newman's theorem, to prove that $\mathcal{R}'$ is confluent, we only need to prove that all critical pairs can be closed. If $A$ is atomic, $A \to_{\mathcal{R}'} B$ and $A \to_{\mathcal{R}'} C$, then there exists propositions $b$ and $c$ such that $A \to_{\mathcal{R}} b$, $A \to_{\mathcal{R}} c$, $B = b''$ and $C = c''$. As the system $\mathcal{R}$ is confluent there exists a proposition $d$ such that $b \to_{\mathcal{R}} d$ and $c \to_{\mathcal{R}} d$. We have $B \to_{\mathcal{R}'} d''$ and $C \to_{\mathcal{R}'} d''$. □

**Proposition 4.4** *If a sequent $A_1, \ldots, A_n \vdash_{\equiv} B_1, \ldots, B_p$ is provable in classical sequent calculus, then $A'_1, \ldots, A'_n, \neg B''_1, \ldots, \neg B''_p \vdash_{\equiv}$ is provable in intuitionistic sequent calculus.*

*Proof.* The proof is an easy induction on the structure of the proof of

$$A_1, \ldots, A_n \vdash_{\equiv} B_1, \ldots, B_p.$$

As an example, let us detail one case. If the last rule is $\vee$-right

$$\dfrac{\dfrac{\pi}{A_1 \ldots A_n \vdash_{\equiv} C D B_2 \ldots B_p}}{A_1 \ldots A_n \vdash_{\equiv} B_1 \ldots B_p E} \vee\text{-right}$$

Where $E \equiv C \vee D$. By induction hypothesis we have an intuitionistic proof $\pi'$ of

$$A'_1 \ldots A'_n \neg B''_2 \ldots \neg B''_p \neg C'' \neg D'' \vdash_{\equiv} .$$

We have $E'' \equiv (C \vee D)'' = C' \vee D' = \neg\neg C'' \vee \neg\neg D''$. We build the following proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\pi'}{A'_1 \ldots A'_n \neg B''_1 \ldots \neg B''_p \neg C'' \neg D'' \vdash_{\equiv}}}{A'_1 \ldots A'_n \neg B''_1 \ldots \neg B''_p \neg C'' \vdash_{\equiv} \neg\neg D''} \neg\text{-right}}{A'_1 \ldots A'_n \neg B''_1 \ldots \neg B''_p \neg C'' \vdash_{\equiv} \neg\neg C'' \vee \neg\neg D''} \vee\text{-right}}{A'_1 \ldots A'_n \neg B''_1 \ldots \neg B''_p \neg(\neg\neg C'' \vee \neg\neg D'') \neg C'' \vdash_{\equiv}} \neg\text{-left}}{A'_1 \ldots A'_n \neg B''_1 \ldots \neg B''_p \neg(\neg\neg C'' \vee \neg\neg D'') \vdash_{\equiv} \neg\neg C''} \neg\text{-right}}{A'_1 \ldots A'_n \neg B''_1 \ldots \neg B''_p \neg(\neg\neg C'' \vee \neg\neg D'') \vdash_{\equiv} (\neg\neg C'' \vee \neg\neg D'')} \vee\text{-right}}{A'_1 \ldots A'_n \neg B''_1 \ldots \neg B''_p \neg(\neg\neg C'' \vee \neg\neg D'') \neg(\neg\neg C'' \vee \neg\neg D'') \vdash_{\equiv}} \neg\text{-left}}{A'_1 \ldots A'_n \neg B''_1 \ldots \neg B''_p \neg E'' \vdash_{\equiv}} \text{contraction}$$

□

**Definition 4.2** *An intuitionistic sequent* $\Gamma \vdash_{\equiv} \Delta$ *is said to represent a classical sequent* $A_1 \dots A_n \vdash_{\equiv} B_1 \dots B_p$ *if each proposition of* $\Gamma$ *has the form* $A_i'', A_i' or \neg B_i''$ *and the proposition of* $\Delta$ *has the form* $B_i', B_i'' or \neg A_i''$.

**Proposition 4.5** *If an intuitionistic sequent* $\Gamma \vdash_{\equiv} \Delta$ *represents a classical sequent* $A_1 \dots A_n \vdash_{\equiv} B_1 \dots B_p$ *and is provable in the cut free intuitionistic sequent calculus modulo* $\mathcal{R}'$*, then the sequent* $A_1 \dots A_n \vdash_{\equiv} B_1 \dots B_p$ *is provable in the cut free classical sequent calculus modulo* $\mathcal{R}$.

*Proof.*

- If the last rule is a structural rule or a logical rule applied to a proposition of $\Gamma$ that has the form $A_i'$ or $\neg B_i''$ or to a proposition of $\Delta$ that has the form $B_i'$ or $\neg A_i''$ then the sequent obtained also represents the sequent $A_1 \dots A_n \vdash_{\equiv} B_1 \dots B_p$. Thus we simply apply the induction hypothesis.
- If the last rule is a logical rule applied to a proposition of the form $A_i''$ or $B_i''$ leading to a sequent $\Gamma' \vdash_{\equiv} \Delta'$, then we duplicate (with the contraction rule) the proposition $A_i$ or $B_i$ in the sequent $A_1 \dots A_n \vdash_{\equiv} B_1 \dots B_p$ and we apply the same rule to this proposition. The obtained sequent is represented by $\Gamma' \vdash_{\equiv} \Delta'$ hence we apply the induction hypothesis.
- If the last rule is an axiom, then we apply an axiom rule.

□

**Theorem 4.1** *If the rewrite system* $\mathcal{R}'$ *has a pre-model then the cut rule is redundant in the classical sequent calculus modulo* $\mathcal{R}$.

*Proof.* If the system $\mathcal{R}'$ has a pre-model then the intuitionistic sequent calculus modulo $\mathcal{R}'$ has the cut elimination property. Let $\Gamma \vdash_{\equiv} \Delta$ be a sequent and $\Gamma' \vdash_{\equiv} \Delta'$ its double negation.

If $\Gamma \vdash_{\equiv} \Delta$ has a proof in the classical sequent calculus modulo $\mathcal{R}$, then $\Gamma' \vdash_{\equiv} \Delta'$ has a proof in the intuitionistic sequent calculus modulo $\mathcal{R}'$, $\Gamma' \vdash_{\equiv} \Delta'$ has a cut free proof in the intuitionistic sequent calculus modulo $\mathcal{R}'$ and $\Gamma \vdash_{\equiv} \Delta$ has a cut free proof in the classical sequent calculus modulo $\mathcal{R}$. □

**Corollary 4.3**

- *The classical sequent calculus modulo the rules of higher-order logic has the cut elimination property.*

- *The classical sequent calculus modulo a confluent and terminating quantifier-free rewrite system has the cut elimination property.*
- *The classical sequent calculus modulo any confluent and terminating positive rewrite system has the cut elimination property.*

*Proof.* The $\neg\neg$-translation of the rewrite system of higher-order logic is that of proposition 3.4. The $\neg\neg$-translation of a quantifier-free rewrite system is a quantifier-free rewrite system. The $\neg\neg$-translation of a positive rewrite system is a positive rewrite system. □

## Conclusion

We have defined generically a wide range of deductive systems. Every formalism is defined by a given rewrite system over first-order propositions. We have seen that the systems so defined go further than first-order logic.

We conjecture that simple combinatorial conditions on the rewrite system imply the cut elimination property and thus logical consistency. This conjecture implies the consistency of higher-order logic. It is also interesting to remark that, provided this conjecture holds, its logical strength is not yet clear. In other words, we do not know what is the strongest logical system definable in deduction modulo a confluent and terminating rewrite system. We have seen though, that naive attempts to encode set theory do not succeed.

In any case, it seems that studying rewrite systems from the point of view of their logical properties is a new, promising and interesting subject.

## References

[1] S. Boutin, Using reflection to build efficient and certified decision procedures. TACS '97, LNCS 1281 (1997).

[2] T. Coquand and G. Huet, The Calculus of constructions, *Information and Computation*, 76 (1988) pp. 95-120.

[3] G. Dowek, Th. Hardin, and C. Kirchner, Theorem proving modulo, *Rapport de Recherche INRIA* 3400 (1998).

[4] J. Ekman, Normal proofs in set theory, *Doctoral thesis*, Chalmers university of technology and University of Göteborg (1994).

[5] H.B. Enderton, A mathematical introduction to logic, *Academic Press* (1972).

[6] M. Fay, First-order unification in an equational theory, *Fourth Workshop on Automated Deduction* (1979), pp. 161-167.

[7] J. Gallier, Logic in computer science, *Harper and Row* (1986).

[8] J.Y. Girard, Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieur, *Thèse de Doctorat d'État*, Université de Paris VII (1972).

[9] J.Y. Girard, Y. Lafont, and P. Taylor. Proofs and Types, *Cambridge University Press* (1989).

[10] L. Hallnäs, On normalization of proofs in set theory, *Doctoral thesis*, University of Stockholm (1983).

[11] J.-M. Hullot, Canonical forms and unification, W. Bibel and R. Kowalski (Eds.) *Conference on Automated Deduction*, Lecture Notes in Computer Science 87, Springer-Verlag (1980), pp. 318-334.

[12] J.L. Krivine and M. Parigot, Programming with proofs, *J. Inf. Process. Cybern. EIK* 26 (1990), pp. 149-167.

[13] P. Martin-Löf, Intuitionistic type theory, *Bibliopolis*, Napoli (1984).

[14] Ch. Paulin-Mohring, Inductive definitions in the system COQ, Rules and Properties, *Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science 664 (1993) pp. 328-345.

[15] G. Plotkin, Building-in equational theories *Machine Intelligence*, 7 (1972), pp. 73–90

[16] M. Stickel, Automated deduction by theory resolution, *Journal of Automated Reasoning*, 4, 1 (1985), pp. 285-289.

[17] W.W. Tait, Intensional interpretation of functionals of finite type I, *Journal of Symbolic Logic*, 32, 2 (1967) pp. 198-212.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399